

CLAIMS

What is claimed is:

5

1. A method for managing objects in a data processing system, the method comprising:

creating a skeleton cache; and

storing a first object in the skeleton cache, wherein a

10 skeleton cache stores skeleton objects and/or full objects, wherein a full object is an object in which each attribute within the full object has a data value, wherein a skeleton object is an object in which at least one attribute within the skeleton object is dataless, and wherein a skeleton

15 object has a corresponding full object that is stored without the skeleton cache but within the data processing system.

2. The method of claim 1 further comprising:

20 retrieving a skeleton definition associated with the first object, wherein a skeleton definition is associated with an first object's type, wherein a skeleton definition indicates whether an attribute within the first object is a skeleton attribute, and wherein a skeleton attribute is a

25 dataless attribute.

3. The method of claim 2 further comprising:
fetching a full object;
generating a skeleton object corresponding to the
fetched full object;

5 determining whether a first attribute in the fetched
full object is a full attribute or a skeleton attribute in
accordance with the fetched full object's skeleton
definition;

10 in response to a determination that the first attribute
is a full attribute, copying the first attribute's data
value from the fetched full object into the corresponding
skeleton object; and

storing the corresponding skeleton object in the
skeleton cache.

15

4. The method of claim 2 further comprising:
requesting a value of a second attribute of a second
object; and

20 determining whether the second object resides in the
skeleton cache as a full object or as a skeleton object.

5. The method of claim 4 further comprising:

25 in response to a determination that the second object
resides in the skeleton cache as a full object, retrieving
the requested value of the second attribute of the second
object from the second object residing in the skeleton
cache.

6. The method of claim 4 further comprising:

30 in response to a determination that the second object
resides in the skeleton cache as a skeleton object,
determining whether the second attribute in the second
object is a full attribute or a skeleton attribute.

7. The method of claim 6 further comprising:
in response to a determination that the second
attribute is a full attribute, retrieving the requested
5 value of the second attribute of the second object from the
second object residing in the skeleton cache.

8. The method of claim 6 further comprising:
in response to a determination that the second
10 attribute is a skeleton attribute, retrieving the requested
value of the second attribute of the second object from a
corresponding full object for the second object stored
within the data processing system.

15 9. The method of claim 1 further comprising:
determining whether to create a skeleton cache in
accordance with a skeleton policy, wherein the skeleton
policy comprises one or more configurable conditions for
determining whether to create a skeleton cache.

20 10. The method of claim 9 wherein a configurable condition
is based upon an identity of a user of the data processing
system.

25 11. The method of claim 9 wherein a configurable condition
is based upon a determination of membership of a user of the
data processing system within a class of users.

30 12. The method of claim 9 wherein a configurable condition
is based upon a determination of membership of a device for
storing the skeleton cache within a class of devices.

13. The method of claim 9 wherein a configurable condition is based upon a determination of available memory.

14. The method of claim 9 wherein a configurable condition 5 is based upon a determination of available memory within a device that stores the skeleton cache.

15. The method of claim 9 wherein a configurable condition is based upon a determination of available network 10 bandwidth.

16. The method of claim 9 wherein a configurable condition is based upon a temporal evaluation of historical updates of objects stored within the data processing system.

15

17. The method of claim 1 further comprising:
providing a database access interface component between a data requester and a database; and

20 embedding a skeleton handler within the database access interface component, wherein a skeleton handler performs operations on the skeleton cache on behalf of the database interface component.

25 18. The method of claim 17 further comprising:
configuring a skeleton policy for the database access interface component, wherein the skeleton policy comprises one or more configurable conditions for determining whether to create a skeleton cache.

30

19. An apparatus for managing objects in a data processing system, the apparatus comprising:

means for creating a skeleton cache; and

means for storing a first object in the skeleton cache,

5 wherein a skeleton cache stores skeleton objects and/or full objects, wherein a full object is an object in which each attribute within the full object has a data value, wherein a skeleton object is an object in which at least one attribute within the skeleton object is dataless, and wherein a
10 skeleton object has a corresponding full object that is stored without the skeleton cache but within the data processing system.

20. The apparatus of claim 19 further comprising:

15 means for retrieving a skeleton definition associated with the first object, wherein a skeleton definition is associated with an first object's type, wherein a skeleton definition indicates whether an attribute within the first object is a skeleton attribute, and wherein a skeleton
20 attribute is a dataless attribute.

21. The apparatus of claim 20 further comprising:

means for fetching a full object;

means for generating a skeleton object corresponding to

25 the fetched full object;

means for determining whether a first attribute in the fetched full object is a full attribute or a skeleton attribute in accordance with the fetched full object's skeleton definition;

30 means for copying the first attribute's data value from the fetched full object into the corresponding skeleton object in response to a determination that the first attribute is a full attribute; and

means for storing the corresponding skeleton object in the skeleton cache.

22. The apparatus of claim 20 further comprising:

5 means for requesting a value of a second attribute of a second object; and

means for determining whether the second object resides in the skeleton cache as a full object or as a skeleton object.

10

23. The apparatus of claim 22 further comprising:

means for retrieving the requested value of the second attribute of the second object from the second object residing in the skeleton cache in response to a

15 determination that the second object resides in the skeleton cache as a full object.

24. The apparatus of claim 22 further comprising:

means for determining whether the second attribute in the second object is a full attribute or a skeleton attribute in response to a determination that the second object resides in the skeleton cache as a skeleton object.

25. The apparatus of claim 24 further comprising:

25 means for retrieving the requested value of the second attribute of the second object from the second object residing in the skeleton cache in response to a determination that the second attribute is a full attribute.

26. The apparatus of claim 24 further comprising:
means for retrieving the requested value of the second
attribute of the second object from a corresponding full
object for the second object stored within the data
5 processing system in response to a determination that the
second attribute is a skeleton attribute.

27. The apparatus of claim 19 further comprising:
means for determining whether to create a skeleton
10 cache in accordance with a skeleton policy, wherein the
skeleton policy comprises one or more configurable
conditions for determining whether to create a skeleton
cache.

15 28. The apparatus of claim 27 wherein a configurable
condition is based upon an identity of a user of the data
processing system.

20 29. The apparatus of claim 27 wherein a configurable
condition is based upon a determination of membership of a
user of the data processing system within a class of users.

25 30. The apparatus of claim 27 wherein a configurable
condition is based upon a determination of membership of a
device for storing the skeleton cache within a class of
devices.

30 31. The apparatus of claim 27 wherein a configurable
condition is based upon a determination of available memory.

32. The apparatus of claim 27 wherein a configurable
condition is based upon a determination of available memory
within a device that stores the skeleton cache.

33. The apparatus of claim 27 wherein a configurable condition is based upon a determination of available network bandwidth.

5

34. The apparatus of claim 27 wherein a configurable condition is based upon a temporal evaluation of historical updates of objects stored within the data processing system.

10 35. The apparatus of claim 19 further comprising:
means for providing a database access interface component between a data requester and a database; and
means for embedding a skeleton handler within the database access interface component, wherein a skeleton
15 handler performs operations on the skeleton cache on behalf of the database interface component.

36. The apparatus of claim 35 further comprising:
means for configuring a skeleton policy for the
20 database access interface component, wherein the skeleton policy comprises one or more configurable conditions for determining whether to create a skeleton cache.

TOP SECRET

37. A computer program product on a computer readable medium for managing objects in a data processing system, the computer program product comprising:

instructions for creating a skeleton cache; and

5 instructions for storing a first object in the skeleton cache, wherein a skeleton cache stores skeleton objects and/or full objects, wherein a full object is an object in which each attribute within the full object has a data value, wherein a skeleton object is an object in which at least one attribute within the skeleton object is dataless, and wherein a skeleton object has a corresponding full object that is stored without the skeleton cache but within the data processing system.

15 38. The computer program product of claim 37 further comprising:

instructions for retrieving a skeleton definition associated with the first object, wherein a skeleton definition is associated with an first object's type,

20 wherein a skeleton definition indicates whether an attribute within the first object is a skeleton attribute, and wherein a skeleton attribute is a dataless attribute.

39. The computer program product of claim 38 further 25 comprising:

instructions for fetching a full object;

instructions for generating a skeleton object corresponding to the fetched full object;

30 instructions for determining whether a first attribute in the fetched full object is a full attribute or a skeleton attribute in accordance with the fetched full object's skeleton definition;

instructions for copying the first attribute's data value from the fetched full object into the corresponding skeleton object in response to a determination that the first attribute is a full attribute; and

5 instructions for storing the corresponding skeleton object in the skeleton cache.

40. The computer program product of claim 37 further comprising:

10 instructions for determining whether to create a skeleton cache in accordance with a skeleton policy, wherein the skeleton policy comprises one or more configurable conditions for determining whether to create a skeleton cache.

15